SPECTRUM®

# AutoDiscovery User's Guide

CABLETRON SYSTEMS

The Complete Networking Solution™

# Notice

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.

## Virus Disclaimer

Cabletron has tested its software with current virus checking technologies. However, because no anti-virus system is 100% reliable, we strongly caution you to write protect and then verify that the Licensed Software, prior to installing it, is virus-free with an anti-virus system in which you have confidence.

Cabletron Systems makes no representations or warranties to the effect that the Licensed Software is virus-free.

Copyright © April 1998, by Cabletron Systems, Inc. All rights reserved.

Printed in the United States of America.

Order Number: 9030727 E12

Cabletron Systems, Inc.
P.O. Box 5005
Rochester, NH 03866-5005

**SPECTRUM**, **SPECTRUM IMT/VNM** logo, **DCM**, **IMT** and **VNM** are registered trademarks, and **AutoDiscovery**, **SpectroGRAPH**, **SpectroSERVER**, **Device Communications Manager, Inductive Modeling Technology**, **Device Communications Manager,** and **Virtual Network Machine** are trademarks of Cabletron Systems, Inc.

**UNIX** is a trademark of UNIX System Laboratories, Inc.
**Ethernet** is a trademark of Xerox Corporation.
**Solaris** is a registered trademark of Sun MicroSystems, Inc.
**Windows** and **Windows NT** are trademarks of Microsoft Corporation

# Restricted Rights Notice

(Applicable to licenses to the United States Government only.)

1. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

   Cabletron Systems, Inc., 35 Industrial Way, Rochester, New Hampshire 03866-5005.

2. (a) This computer software is submitted with restricted rights. It may not be used, reproduced, or disclosed by the Government except as provided in paragraph (b) of this Notice or as otherwise expressly stated in the contract.

   (b) This computer software may be:

   (1) Used or copied for use in or with the computer or computers for which it was acquired, including use at any Government installation to which such computer or computers may be transferred;

   (2) Used or copied for use in a backup computer if any computer for which it was acquired is inoperative;

   (3) Reproduced for safekeeping (archives) or backup purposes;

   (4) Modified, adapted, or combined with other computer software, provided that the modified, combined, or adapted portions of the derivative software incorporating restricted computer software are made subject to the same restricted rights;

   (5) Disclosed to and reproduced for use by support service contractors in accordance with subparagraphs (b) (1) through (4) of this clause, provided the Government makes such disclosure or reproduction subject to these restricted rights; and

   (6) Used or copied for use in or transferred to a replacement computer.

   (c) Notwithstanding the foregoing, if this computer software is published copyrighted computer software, it is licensed to the Government, without disclosure prohibitions, with the minimum rights set forth in paragraph (b) of this clause.

   (d) Any other rights or limitations regarding the use, duplication, or disclosure of this computer software are to be expressly stated in, or incorporated in, the contract.

   (e) This Notice shall be marked on any reproduction of this computer software, in whole or in part.

# Contents

# Preface

*This document provides instructions for using AutoDiscovery, a SPECTRUM core application program that operates in conjunction with the SPECTRUM network management software to create a model of an existing network.*

## Who Should Read This Guide

This guide is intended for SPECTRUM administrators and technicians responsible for determining network configuration and overseeing network operations. By following the instructions and procedures described herein, the reader will be able to use AutoDiscovery to create and maintain an accurate network model that will facilitate full exploitation of SPECTRUM's powerful management and monitoring capabilities.

This guide assumes that the administrative user is experienced with SPECTRUM and its administration as described in the SPECTRUM Administration documentation. Any user should also be familiar with the SpectroGRAPH™ user interface, and with the user functions explained in the SPECTRUM Operation documentation.

## How to Use This Guide

This guide contains both general information and detailed instructions. The document is organized as follows:

- Chapter 1 provides an overview of AutoDiscovery and describes the process it uses to construct a network model. This chapter also identifies the limitations and constraints within which the application functions.
- Chapter 2 explains how to use the AutoDiscovery dialog box to establish network bounds and other settings for a discovery session.

- Chapter 3 provides step-by-step instructions for running AutoDiscovery and suggests strategies for configuring your network topology views after discovery operations are completed.
- Appendix A discusses IP address conventions and subnetting schemes.
- Appendix B explains how to run AutoDiscovery via UNIX crontab scripts. This lets you take advantage of automatic discovery capabilities on a continuous or periodic basis.
- Appendix C is a glossary of the technical terms and acronyms used in this guide.

# Questions about SPECTRUM Documentation


Send your questions, comments or suggestions regarding SPECTRUM documentation to the Technical Communications Department directly via the following internet address:

**spectrum-techdocs@ctron.com**

# Chapter 1

# The Discovery Process

*This chapter provides an overview of AutoDiscovery and describes the methodology it uses to construct a network model. Constraints and limitations that apply to the process are identified along with certain variables that can affect the duration of a discovery session.*

## What is AutoDiscovery?

AutoDiscovery is a key part of SPECTRUM. It is an application program that virtually automates the creation and maintenance of the software model SPECTRUM uses to manage your network. Operating within IP address ranges and other guidelines that you specify, AutoDiscovery explores your network and creates individual models of the devices and other network entities it finds. These models are stored in the VNM database along with information about their relationships and interconnections. SPECTRUM's graphical user interface, SpectroGRAPH, provides access to the overall network model through a hierarchy of Topology views in which AutoDiscovery places icons representing each of the network elements that are modeled in the database.

## The Benefits of Using AutoDiscovery

By automating most of the modeling process, AutoDiscovery can save significant amounts of time for SPECTRUM users, especially in the case of very large networks. However, beyond the time saved in finding devices, creating models, and placing icons in the topological representation of your network, AutoDiscovery can enhance the effectiveness of SPECTRUM itself. For one reason, AutoDiscovery performs a very comprehensive exploration; it will discover elements you might well miss if attempting to manually model your network. Furthermore, the network model AutoDiscovery creates is specifically designed to accommodate the particular way that SPECTRUM works. AutoDiscovery's placement of models within the topological scheme

supports SPECTRUM intelligence and optimizes its ability to identify problem areas and accurately reflect current conditions.

The end result of the AutoDiscovery process is a stratified abstraction of your network that lets you access information at the precise level of detail you need. This makes it easier for you to make critical management decisions whether on a high-level, network-to-network basis, or down at the hub port or endpoint device level.

# Modeling and Mapping Conventions

As AutoDiscovery "discovers" devices and other network entities, it creates the appropriate SPECTRUM models and adds them to the SpectroSERVER database. AutoDiscovery also creates icons to represent these entities (and the logical connections* between them) and by default places the icons in the appropriate SPECTRUM Topology views according to the rules described below. Some of these defaults can be overridden through the Discovery Options described on Page 1-7.

1. Icons representing routers that connect two or more IP Class A, B or C networks are placed at the Universe (top) level of the SPECTRUM topology scheme.

2. Icons representing IP Class A, B and C network models are placed at the Universe level.

3. If a router connects subnets within an IP Class A, B, or C network, the router icon is placed within the Topology view for that network's model. Each subnet connected by such a router is modeled as a LAN, and the corresponding icon is placed within the same view. Note that a single LAN model may embrace multiple subnet ranges. AutoDiscovery automatically combines multiple addresses into a single LAN when routers are configured with multiple addresses on an interface. Note also that if a router within an IP Class A, B, or C network has only a single non-serial interface, it will be modeled within the LAN associated with that interface. WA_Link models are created for wide-area type interfaces (T1, T3, X.25, etc.) If a wide-area interface has multiple subnets associated with it, a separate WA_Link model will be created to represent each logical connection.

4. Icons for bridges and for the discrete LANs they interconnect are placed within the Topology view for the model of the LAN that they comprise. (Discrete LAN model types include: 802.3, 802.5, FDDI, etc.)

5. Icons for hub and fanout models are placed within the Topology view of the discrete LAN model to which they belong. The fanout model type is used to model any non-intelligent or unidentifiable device that provides connectivity within a LAN. Examples of fanouts include coaxial cable segments, media access units (MAUs), and multiport transceivers.

6. Endpoint device icons are connected to the appropriate port within a Device Topology view, if the device's address is the **only** one heard on that port. Icons for endpoint devices connected to fanouts, are placed within the fanout model's Cablewalk view or Cablewalk List view.

* By default, AutoDiscovery creates gold or silver "pipe" icons to indicate logical connections between network entities. A connection is said to be "resolved" when SPECTRUM can determine specific device ports at both ends. The pipe icon is colored gold when a connection is resolved, silver when the connection has not been resolved. For example, if AutoDiscovery places a gold connection pipe between a router model icon and a LAN model icon, one of the ports in the router's DevTop view will show a connection to the LAN and to some other device within that LAN. Likewise, the other device's DevTop view will show a connection to the router (and to the network group entity that contains the router). You can also draw connections between icons manually, and SPECTRUM will attempt to resolve them. However, unresolved user-drawn connections may be erased by subsequent AutoDiscovery sessions if the connections do not comply with the modeling and mapping conventions followed by SPECTRUM.

SPECTRUM 5.0 also features a resource-conserving Live Pipes service that you can enable/disable for the VNM (server) model as a whole, then toggle on and off as desired for selected connections. When this service is enabled, a different range of pipe colors is used to indicate the current status of the link (good, bad, disabled, unknown, or unreachable). For further information and a key to the Live Pipes status color code, refer to ***Getting Started for Administrators*** or ***How to Manage Your Network with SPECTRUM***.

# Phases of Discovery

As the modeling and mapping conventions discussed in the previous section suggest, AutoDiscovery is designed to operate in a hierarchical manner. That is, it explores the network and populates SPECTRUM Topology views from the top down. The largest or most general network groups are placed within the Universe level Topology view with subnets and devices appearing in successively lower level views until, finally, endpoint devices are shown connected to specific ports in DevTop views or to cable segments in Cablewalk views.

The scope of this process depends upon the IP address ranges and other guidelines you establish using the main AutoDiscovery dialog box described in Chapter 2. It also depends on the particular level of the topological hierarchy from which you are running the AutoDiscovery application. When you start at the Universe level using all of the available discovery methods and protocols (and a sufficiently broad address range), discovery proceeds automatically, and the entire network can be explored and mapped in a single session. In practice, however, and especially for larger networks, it is often preferable to first run the application at the Universe level with only the Router Discovery

method selected, so that discovery proceeds only to the level governed by Rules 1 through 3 (i.e., mapping only routers, IP Class A, B, or C networks, LANs and wide area links). AutoDiscovery can then be run separately for each of the discovered LANs. This approach allows the administrator to more closely monitor the process and facilitates isolation and resolution of any anomalies or problems encountered. In any case, the complete discovery process occurs in three basic phases, which are described in the following sections as if they are occurring in a single session.

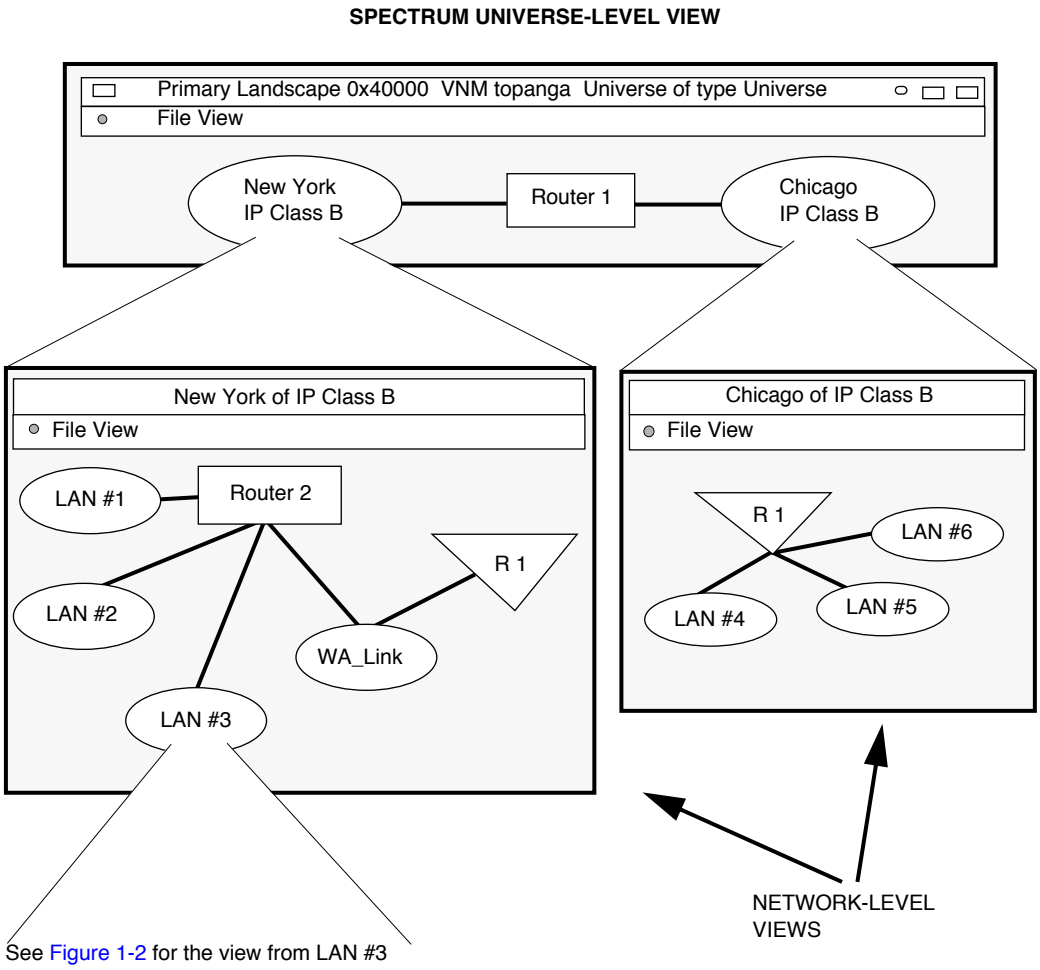
## Phase One - The Router/Network Level

AutoDiscovery always begins by checking the SPECTRUM database for existing models of "seed" routers whose route tables can be used as a source of information to "grow" the initial high-level topology. For all such routers that are currently responding (and whose models are not currently in SPECTRUM's Lost and Found View), AutoDiscovery reads the route table to identify other routers listed as **next hops**. The next hop addresses are examined in terms of the search range established for the discovery session. Those that are within the range (or that are associated with **destination** addresses within the range) are added to the list of routers that will be processed in this same manner. If no model currently exists for the next hop address, AutoDiscovery creates one. Subnets that the table indicates are associated with a particular interface are stored in the LAN model associated with that interface. Again, AutoDiscovery creates the LAN model if it doesn't already exist.

As routers on the list are being processed, AutoDiscovery creates the appropriate network models (IP Class A, B, or C) and populates them with models of the discovered LANs and wide area links, placing router models at the same topological level as the entities they connect. Placement of models is accomplished using the interface and network mask and class information obtained from the route tables and is done in accordance with the modeling and mapping conventions previously described.

Figure 1-1 is an example of the topology mapping that might result from Phase One discovery. The Universe-level Topology View in the top part of the figure shows two IP Class B networks connected by a router. This router is shown at the top level because it is a border router (i.e., one that connects two or more distinct IP networks). By opening the Topology View for either of the Class B network icons (New York or Chicago), you can view the contents of these networks as shown in the lower portion of the figure. The view on the left contains icons representing three LANs connected via an interior router called Router 2. This router is connected by a wide area link back to Router 1, which is represented by a triangular off-page reference icon. Likewise, the view on the right shows Router 1 as an off-page reference icon. Since Router 1 actually connects the two Class B networks, it is more accurate to show its device icon at the Universe level between the network icons rather than inside the views for those networks.

**Figure 1-1. Typical Phase One Discovery Results**




## Phase Two - The LAN/Bridge Level

The second phase of discovery occurs at the LAN/bridge level. Here AutoDiscovery uses one or more of three user-selectable discovery methods* to examine each of the LANs discovered in Phase One. In the course of examining each LAN, AutoDiscovery locates and models all supported bridges and uses their interface information to model and place discrete LANs (802.3, 802.5, etc.) that the bridges interconnect. If no bridges are found, hubs will be examined, and if any hub shows an interface type corresponding to a discrete

LAN model type, then a discrete LAN model will be created. The discrete LAN model will collect all hubs showing the same interface type if they are not already collected by another discrete LAN model.

The top part of Figure 1-2 shows the results of Phase Two discovery for LAN #3, which was part of the "New York" IP Class B network-level view shown in Figure 1-1. This view shows that LAN #3 contains three 802.3 discrete networks, one 802.5 discrete network, and two bridges. There is also an off-page reference icon to show the connection between LAN 802.3 #2 and Router 2, whose device icon was placed in the New York network view in Figure 1-1.

*The Range-Test, NIS, and Address Resolution Table discovery methods are selectable from the main AutoDiscovery dialog box, which is discussed in detail in Chapter 2 of this guide.

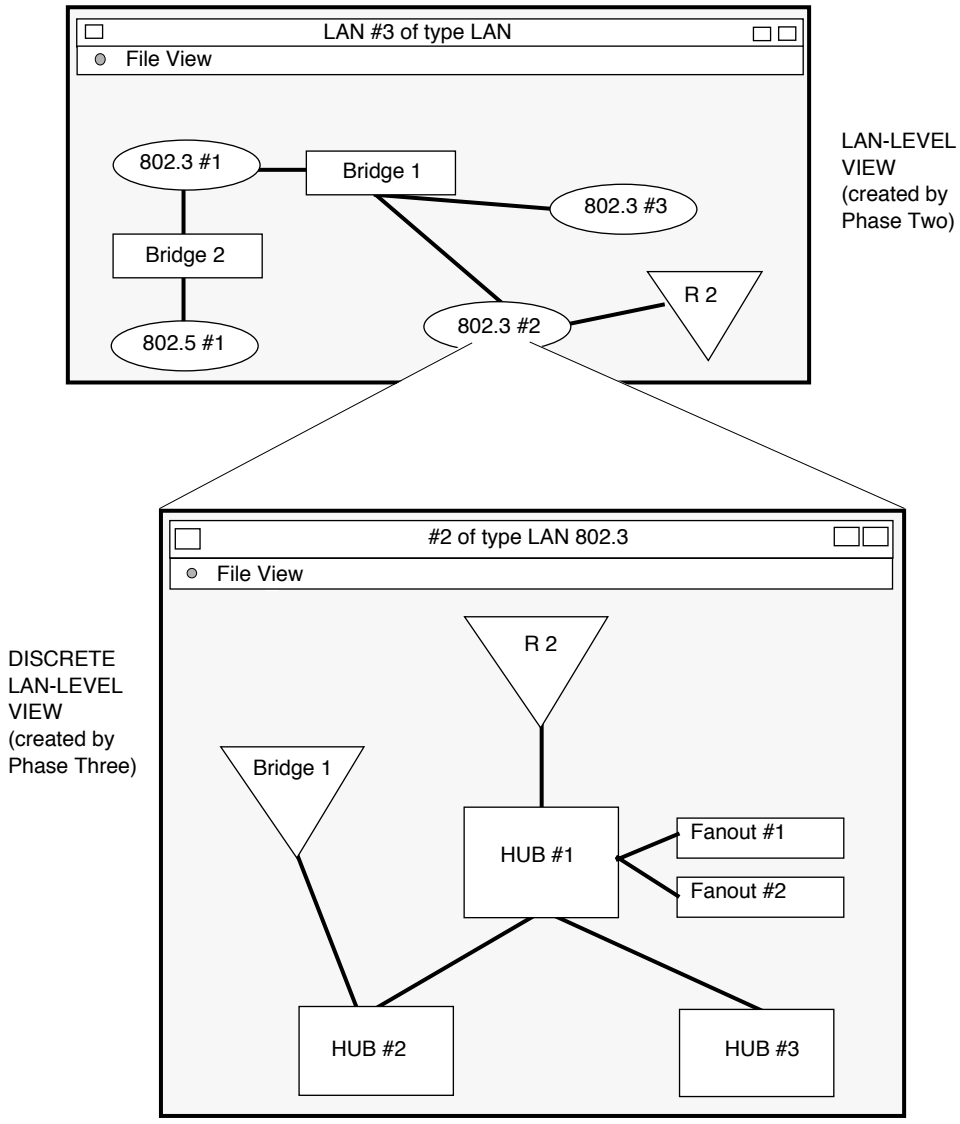
## Phase Three - The Discrete LAN/Hub Level

In the third phase of discovery, the selected discovery methods are used to examine each of the discrete LANs discovered in Phase Two. As each hub is located and modeled, AutoDiscovery also attempts to identify and model the devices connected to each of its ports. (As noted in Item 6 under "Modeling and Mapping Conventions," a model will be placed on a hub or bridge port only if the model's address is the only one heard on the port.) Other non-intelligent (or unidentifiable) multiport devices, such as a multiport transceiver with several users attached, are modeled as fanouts.

The lower view in Figure 1-2 shows the results of Phase Three discovery for the discrete LAN 802.3 #2. Here three hubs and two fanouts have been discovered. Two off-page reference icons also appear in the view, one representing the connection to Bridge 1 at the next higher topological level and the other showing the connection to Router 1, whose device icon appears in Figure 1-1.

As the figures illustrate, when all three phases of discovery have been completed, the result is a comprehensive, topologically accurate representation of your network that lets you access the level of detail you want, from major network groups down to endpoint devices.

**Figure 1-2.** Typical Discovery Results for Phases Two and Three




# Discovery Options

Some of AutoDiscovery's modeling and mapping conventions (see Page 1-2) can be selectively overridden for Universe, Network, and LAN models through the **Discovery Options** panel that appears in the associated Information view. If you are already in the Topology view of the model for which you want to access these options, select **View > Current View Information**. If you are

in the Topology view that *contains* the model, highlight the model's icon, then select **View > Icon Subviews > Model Information**. The location of the Discovery Options panel within the Information view varies depending on which model type is selected, but in each case the panel provides the following three menu buttons, each of which lets you select an alternative to the default way in which models will be created and placed during AutoDiscovery.


The default setting for this button will automatically place routers according to the modeling and mapping conventions described on Page 1-2. Alternatively, you can toggle this button to the **Place Routers in this view only** option, which will place any newly discovered routers in the view from which you invoked AutoDiscovery.


The default setting for this button will create models for IP Class A, B and C networks and place the icons in the Universe Topology view. You can suppress creation of IP Class models by toggling this button to the **Place LANs in this view only** option, which will create LAN models instead of IP Class models and place them only in the view from which you invoked AutoDiscovery.


The default setting for this button will create and place LAN models according to the conventions described on Page 1-2. You can suppress creation of LAN models by toggling this button to the **Don't Create LAN models** option, which will prevent creation of new LAN models during Phase One discovery.

# Duration of Discovery Sessions

The amount of time required to run AutoDiscovery depends on many variables, including:

- The magnitude of the network address range(s) you specify
- The number of discovery methods you choose
- The number of discovery protocols you choose
- The number of community names you specify
- The subnetting scheme in use
- The number of protocols AutoDiscovery must use to identify a device
- The number and type of nodes in your network

Given the number of variables involved, it is impossible to accurately predict discovery time based solely on the number of nodes in a network. The time

needed for a complete network discovery can vary from minutes to hours, depending on the nature of the network involved.

# Constraints

The network map created by AutoDiscovery can only be as accurate as the information available from the network devices. Since this information is typically derived from memory caches within a device, some information may be missing from a device at the time AutoDiscovery contacts it. In other cases, the information available from two different devices may be in conflict, especially right after a change in configuration. Thus you may not discover every device on your network the first time you run AutoDiscovery, and you should consider running additional discovery sessions to further refine your network model with the most recent device data. Indeed, if the configuration of your network changes frequently, AutoDiscovery should be run at regular intervals, either using the Background Discovery feature described in Chapter 2, or as a UNIX cron job as described in Appendix B. Each succeeding session will improve the accuracy and completeness of your model.

Beyond the issue of device status at any given moment, you should keep in mind that AutoDiscovery can only map correctly using repeaters, bridges, and routers for which management protocols provide the necessary port connectivity information (addresses heard on each port). Moreover, the appropriate SPECTRUM management modules must be available to provide access to that information.

You should also be aware that AutoDiscovery cannot *remove* devices from the database nor from the visible topological model, since there is no positive indication that a device has been removed from the network. A device that can not be contacted may be powered off, temporarily out of service, or simply not transmitting. Therefore, if a device is permanently removed (physically) from your network, you must manually destroy the device's model from the database. Otherwise, the model will continue to generate a Lost Contact alarm (condition color red).

The following device-specific constraints and limitations also apply:

- Each router interface to a given network must have the same subnet mask (see Appendix A for more information on subnet addresses).
- Completely redundant routers may not be detected unless either the Range-Test or NIS discovery methods are used (see Chapter 2 for more information on discovery methods).
- If a combination of routing protocols imposes artificial fragmentation on a network, each such fragment should have its own “seed” router to facilitate full discovery of its contents.
- Redundant Cabletron SNMP bridges are supported and will be placed in the proper LANs, but the views will require editing to form adjacencies.

This is because bridges in standby mode have unreliable bridging tables, and AutoDiscovery will not be able to determine the proper port connections.

- Bridges with long timeout periods for their bridging tables may continue to generate information for devices that have since been removed.
- Although IRBMs are not supported as bridges, AutoDiscovery will properly find and place these devices as hubs.
- In spanning tree protocol, a bridge that is a leaf on a spanning tree only broadcasts spanning tree information on the port connected to the "upstream" bridge. Thus a "downstream" hub may never hear any transmissions using the MAC address of an adjacent bridge, and the topology generated by AutoDiscovery may not show the hub-bridge connection.
- Discovery of hubs may generate physical address models in the Lost and Found View.

# Chapter 2

# The User Interface

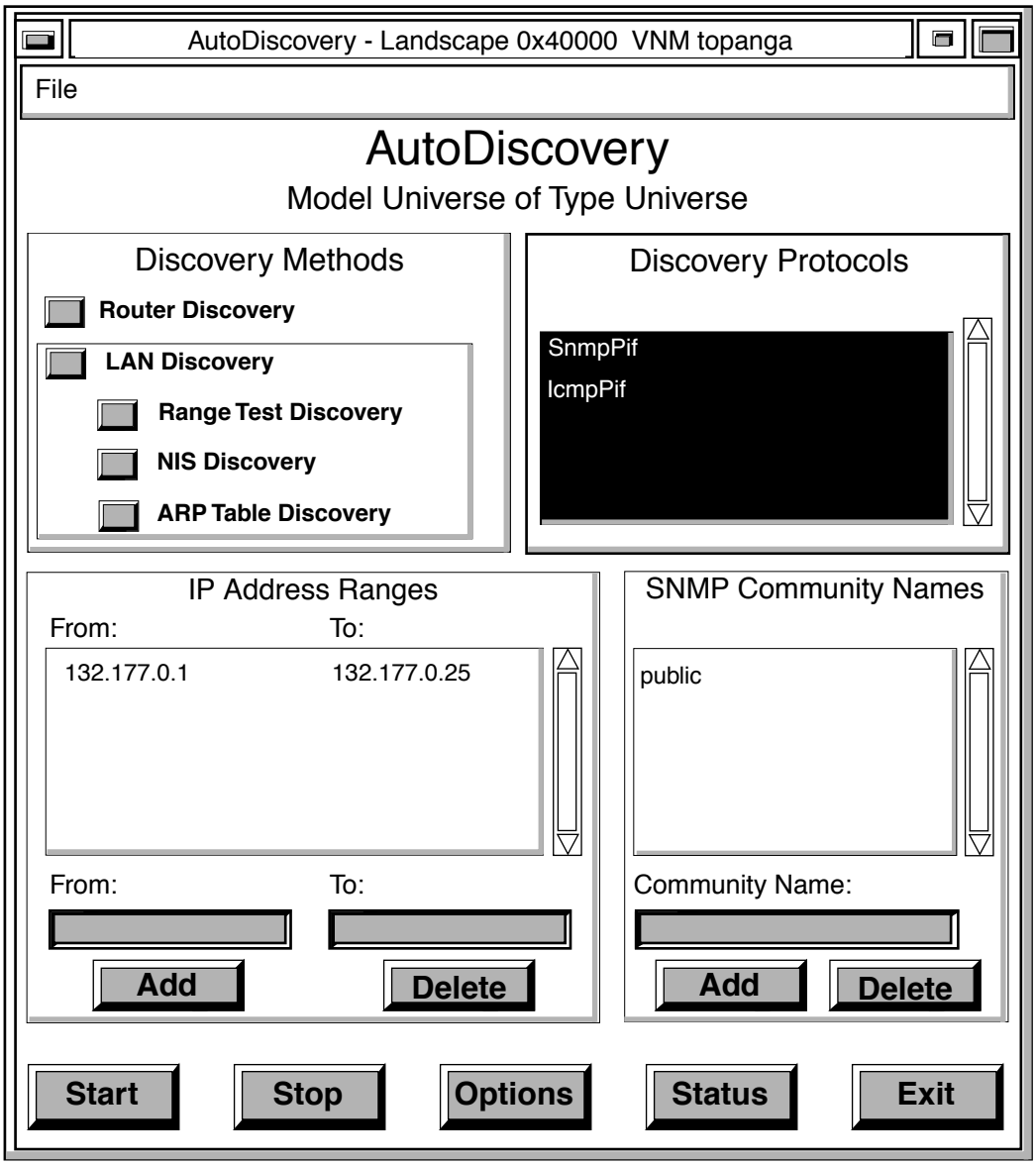
*This chapter explains how to access and operate the dialog boxes that the user interface for the AutoDiscovery application.*

## Accessing AutoDiscovery

AutoDiscovery can be run either from within SpectroGRAPH or from your operating system's command line (see Appendix B for command format). Either method provides access to the user interface through which you determine the boundaries of the network to be discovered and the type of discovery and mapping that will be performed. Once settings are established, you can run subsequent discovery sessions automatically at predetermined intervals using either the *Background Discovery* feature described on Page 2-10, or a UNIX "cron" facility as explained in Appendix B.

To invoke AutoDiscovery from SpectroGRAPH, select the **Auto Discover** option from the **Edit** menu of any SPECTRUM Topology view for which discovery would be applicable (i.e., above the level of discrete LAN). This will display the main AutoDiscovery dialog box shown in Figure 2-1. Note that the title bar identifies the applicable landscape handle and VNM name. There is also a subtitle indicating the Model Name and Model Type of the model from which you invoked AutoDiscovery.

**Figure 2-1.** The AutoDiscovery Dialog Box




# Establishing Discovery Settings

The dialog box shown in Figure 2-1 lets you control and focus the discovery process by determining the discovery method(s) to be used and the protocols and address ranges to be considered. These settings can be saved (via the **File** menu) for default use in subsequent discovery sessions. Individual selection buttons and data entry fields are described on the following pages.

# Discovery Methods

Whenever you run AutoDiscovery, it performs Phase One discovery to identify routers, LANs and wide area links (see Chapter 1 for an overview of the three phases of discovery), thus the Router Discovery button is permanently set to "on" by default. Subsequent actions, however, depend upon the other discovery methods that you select. **The first time you run AutoDiscovery, you should do so at the Universe level with Router Discovery the only method selected.** For subsequent sessions, however, you can activate any of the following three discovery methods by clicking on the associated selector button. Note that if none of these discovery methods is selected, AutoDiscovery will stop after Phase One and will not attempt to discover details for each LAN.

### Router Discovery

When this method is selected, AutoDiscovery searches the SPECTRUM database for a "seed" router model that it can use as the basis for Phase One discovery (see Chapter 1 for an overview of the three phases of discovery). As a default, Router Discovery uses the seed router's IP Routing Table to find and model other routers listed as next hops. Alternatively, you can specify that the router's IP Address Table be used (see *AutoDiscovery Options* on Page 2-8). In addition to router models, Router Discovery creates the appropriate IP Class A, B, or C models and populates them with LAN and WAN models according to the modeling and mapping conventions described in Chapter 1.

### LAN Discovery

This method enables the mapping of existing device models (e.g., those already discovered during a Background Discovery session) at the LAN/bridge level and below. Although you can run AutoDiscovery with only this method selected, it is automatically selected whenever any of the following three methods (Range Test, NIS, or ARP Table) are selected.

### Range Test Discovery

When this method is enabled, AutoDiscovery uses ICMP echo requests (pings) to test each of the IP addresses within the range or ranges you specify in the IP Address Ranges panel explained below. An address that responds to a ping is then tested against selected protocols until it can be identified and the appropriate model created. Although this method provides comprehensive coverage of a given range, the tradeoff in terms of bandwith usage should be considered before using it on larger ranges.

### NIS Discovery (Solaris systems only)

Unless another method is also selected, this method limits discovery to those devices addressed in the host table for your Solaris system's NIS (Network Information Service) server. Although this method is not dependent on any

particular protocol, subsequent identification of devices is facilitated by having both the available Discovery Protocols selected.

### ARP Table Discovery

This method allows AutoDiscovery to associate a discovered device's IP address to a physical (MAC) address so that resolution to the device port level can be achieved duringPhase Three discovery. To do this, AutoDiscovery references the Address Resolution Protocol (ARP) Table of each modeled router and tests any IP addresses within the specified range. For models that already exist in the SpectroSERVER database, the MAC_Address attribute is updated with the physical address associated with the corresponding IP address in the table. If no model exists for a table entry, AutoDiscovery will attempt to create the appropriate model based on the selected protocol(s). For example, if the device does not respond to SnmpPif but does respond to IcmpPif, a pingable model will be created.


The ARP Table method of discovery should not be used as the sole method, since models would not be created for any devices that do not appear in the ARP tables of your routers (such as bridges and hubs). Note also that this method will not produce any results if there are no routers within the specified address range(s).

## IP Address Ranges

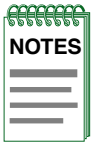
This panel allows you to set the bounds of your network by specifying one or more IP address ranges within which discovery operations will be confined. The top section of the panel (adjacent to the scroll bar) lists the ranges that will be tested when you actually start the application. Each line in the list defines a range by a low address and a high address. For instance, in the example shown by Figure 2-1, AutoDiscovery will test addresses between 132.177.0.1 and 132.177.0.25.

In the lower portion of this panel there are **From** and **To** data entry fields. To add a new range to the list, click within the **From** field on the left and type in the IP address for the low end of the desired range. Then tab to or click within the **To** field on the right and enter the address for the high end. Finally, click on the **Add** button at the bottom of the panel to transfer your entry to the list. To delete a range from the list, click on the desired range to highlight it, then click on the **Delete** button. (Note that the **Delete** button is disabled unless one or more ranges are selected.) To modify a range in the list, double-click on it to move it down to the data entry fields, then edit as needed and add it back to the list.

The first time you run AutoDiscovery at the Universe level, no default ranges will have been established yet, so you will have to enter at least one range. A single range is appropriate if your network's address bounds are contiguous; however, if your network has multiple IP Class A, B, and C networks, you will

need to specify multiple address ranges. Also, if you are sure that your host addresses are limited to a subset of a full network range, discovery will proceed faster if you specify only those ranges currently in use.

You can save the bounds and other settings in this dialog box by pulling down the **File** menu and selecting the **Save Current** option. This will establish the settings as defaults for subsequent executions of AutoDiscovery and is especially useful if you plan to use the Background Discovery feature (see Page 2-10), or to run AutoDiscovery at regular intervals via a UNIX *cron* job (see Appendix B). Note that address range bounds need only be set at the Universe level. If you later invoke AutoDiscovery for a network entity discovered during an initial discovery session (IP network, LAN, wide area link), this panel will show the appropriate bounds. To speed the discovery process, however, you may want to narrow some of these bounds to encompass only the portions of these ranges currently in use.




As noted above, when running AutoDiscovery at lower topological levels, you can specify subsets of the address range(s) used for Universe-level discovery, but you cannot use addresses outside of the range or ranges that have been saved either at the Universe level or for the lower level. In other words, you can scope down but not up. For example, assume the IP address range saved at the Universe level is 132.177.120.1 through 132.177.120.254. If you then try to run AutoDiscovery for a 132.177.0.0 Class B network and enter a range of 132.177.124.1 through 132.177.124.254, you will get an "address out of bounds" error message since the 124 subnet is not within the saved range.

When AutoDiscovery is run from the command line, user-entered ranges for a given LAN will by default override the ranges established for that LAN when AutoDiscovery was run at the Universe level. To prevent the originally established ranges from being further restricted, you can add the argument **`-nolanrestrict`** to the startup command. See Appendix B for further information on command format.


If you are running AutoDiscovery on a network or a part of a network with no routers, be sure that the IP address range(s) you use contain no broadcast addresses, since all devices will respond to the ping, and a device model may be created for whichever one responds first. The resultant model will then regularly poll the broadcast address, which may degrade network performance and have unpredictable results on SPECTRUM operations.

## Discovery Protocols

Select one or both of the discovery protocols listed in this panel by clicking on the protocol name(s). When a protocol is selected it appears highlighted. You can also deselect a highlighted protocol by clicking on it.

AutoDiscovery uses the selected protocols in its attempts to contact and identify a network entity at each IP address in the specified range(s). Protocols are tried in the same order in which they are listed in this panel, and AutoDiscovery categorizes the entity according to the first protocol to which it responds. Thus, a device capable of responding to both SNMP and ICMP protocols will be identified by the SNMP protocol if both are selected.

Each protocol used adds to overall discovery time, so you should try to select *only* those protocols that you actually need for your network. However, if you do not select any protocols, AutoDiscovery will yield a large number of physical address and pingable models, rather than models of the actual devices. For example, if you are using the NIS discovery method, you should select all the available protocols to obtain the best results. The protocols currently available are:

**SnmpPif** Select this protocol to locate SNMP-compliant devices. When this protocol is selected, the SNMP Community Names box is activated. You must specify at least one SNMP community name in this box. The default name is "public."

**IcmpPif** Select this protocol when using the Range-Test method to discover devices that do not support SNMP management but that do support the ICMP echo function.

## SNMP Community Names

Community names are passwords assigned to individual SNMP devices to control access. If you have selected SnmpPif as a discovery protocol, this panel lets you specify one or more SNMP community names that will be searched for during the discovery process. Use the default name "public" if you have not configured your SNMP devices with community names. To add a different name, click within the **Community Names** data entry field, type the name, then click on the **Add** button. To delete a name from the list, click on the name to select it, then click on the **Delete** button. To modify a name in the list, double-click on it to move it down to the data entry field, then edit as needed and add it back to the list. Note that if you delete the community name "public" but have selected the IcmpPif protocol in addition to SnmpPif, then "public" devices will still be discovered and modeled as pingables.

The discovery process searches your network for community names in the order in which they appear in this panel. The first community name a device responds to is the one that is used for the model. If your devices support multiple community names, and you have a preference as to naming, be sure to list names in order of preference.

# File Menu and Control Button Options

With the exception of the **Settings** option, which is accessed only from the AutoDiscovery dialog box's **File** menu, you can select the following options either from the menu or by using the buttons at the bottom of the dialog box.

**Start** — This menu option/button starts AutoDiscovery with the currently displayed settings and brings up the AutoDiscovery Status window shown in Figure 2-3. Note that once AutoDiscovery starts, the main dialog box is disabled except for the **Stop** and **Status** buttons described below.

**Stop** — This menu option/button lets you stop an AutoDiscovery session that is already in progress. Any discovery results to that point will be saved.

**Settings** — This **File** menu option accesses a submenu with the following options.

**Save Current.** Saves all currently specified settings from the AutoDiscovery and AutoDiscovery Options dialog boxes to the VNM database so that you can recall them for future discovery sessions. With the exception of IP address ranges set at the Universe level, these settings apply only to the network model represented by the current view, i.e., the view from which you invoked AutoDiscovery. Settings may differ for different network or LAN models.

**Restore Originals.** Overwrites the current settings with the most recently saved settings recalled from the VNM database.

**Restore Defaults**. Replaces current settings with default settings. There are two sets of default settings: those that apply at the Universe level and those that apply at other levels.

Universe-level defaults include the following: no discovery methods selected, IP Address ranges blank, all protocols selected, SNMP community name of "public."

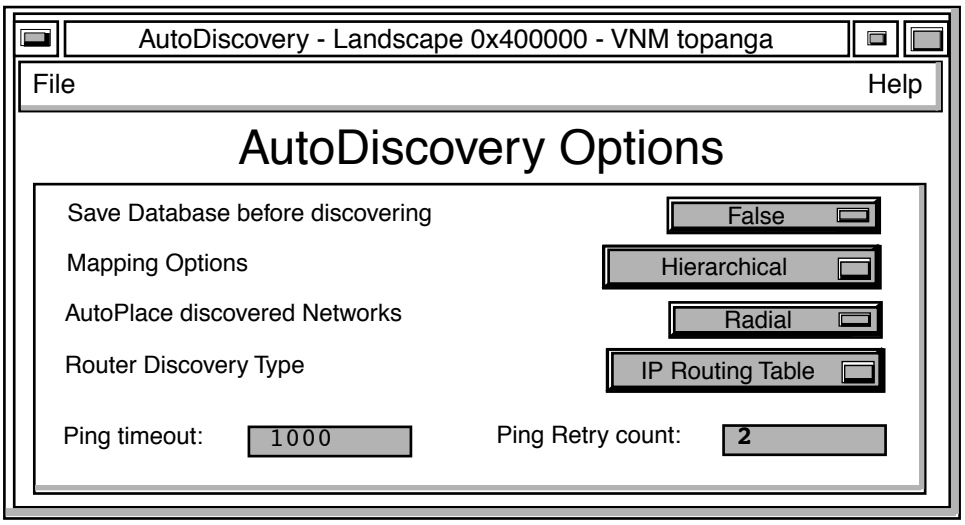
Defaults for lower topology levels are the same as the settings you specified at the Universe level, with the exception of IP Address Ranges. Default ranges at levels other than the Universe level are taken from the IP address list maintained by the model that discovery is run against. As noted previously, however, the IP ranges you specify at the Universe level will bound discovery at all other levels. If a subnet is out of range of Universe-level bounds, it cannot be discovered.

**Options** This menu selection/button displays the AutoDiscovery Options window shown in Figure 2-2.

**Status** This menu selection/button also brings up the AutoDiscovery Status window shown in Figure 2-3.

**Exit** Click on the **Exit** button to close the dialog box without running AutoDiscovery. Both the Options dialog box and the Status window provide a "Close" function as the sole option under their File menus. This option closes the dialog box but does not exit AutoDiscovery.

**Figure 2-2. The AutoDiscovery Options Dialog Box**




## AutoDiscovery Options

This dialog box allows you to further configure and control the AutoDiscovery process through the menu buttons and data entry fields described in the following paragraphs. To dismiss the dialog box, pull down its **File** menu and select the **Close** option.

**Save Database before discovering.** Default setting is **False**. Set the selector to **True** to ensure that a backup copy of your current database is made before discovery operations begin.

**Mapping Options.** The **Hierarchical** option, which is the default, uses a standard IP hierarchical mapping that places models for IP Class A, B, and C networks and the routers connecting them in the Universe-level Topology view, LANs and their connecting routers in a network-level view, discrete LANs in a LAN-level view, and so on. For networks that contain many bridges

and few or no routers, the **Flat** option will place all discovered elements into a fanout model within the current view. The elements can then be viewed through the fanout model's Cablewalk view or Cablewalk List view. This option is mainly intended for use within LANs modeled during Phase One discovery.

**AutoPlace discovered Networks.** Use the **None** setting if you wish to arrange discovered elements manually by dragging them with the mouse once the discovery session is over. In this case, AutoDiscovery will simply cascade the icons in a single overlapping group within the Topology view. The other two settings will automatically arrange the icons in either a **Radial** or **Tree** arrangement once the Topology view is taken out of Edit mode. **Radial** is the default setting.

**Router Discovery Type**. When the Router Discovery method has been selected, this menu button lets you determine whether the discovery process will be based on reading IP Routing Tables or IP Address Tables. Use the default option, **IP Routing Table**, if your primary objective is complete and comprehensive mapping; use the **IP Address Table** option if you are more concerned with reducing the amount of time required.

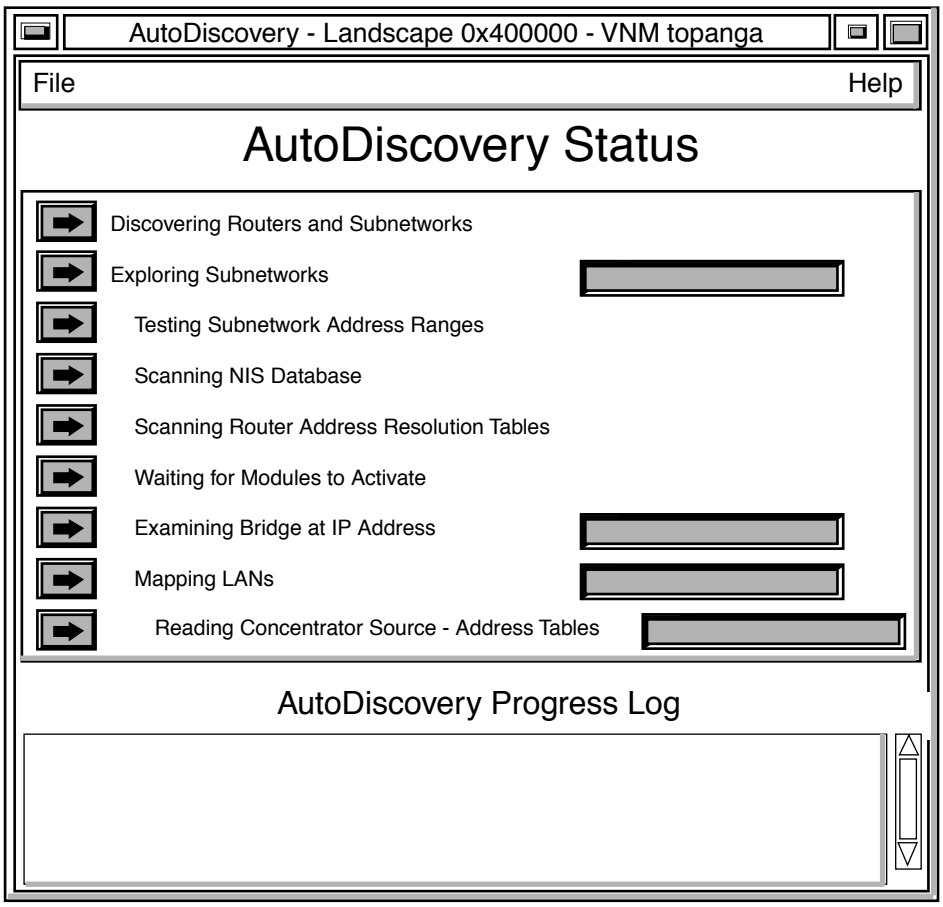
**Ping Timeout**. This field specifies the number of milliseconds per try that AutoDiscovery will wait for an IP address to respond to a ping. The default value is 1000.

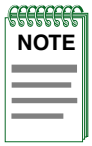
**Ping Retry Count**. The value in this field specifies how many times AutoDiscovery will attempt to ping each IP address. The default is 2.

## AutoDiscovery Status

The AutoDiscovery Status window lets you monitor the progress of the discovery session in two ways. The arrow symbols in the top panel light up (i.e., the arrow appears white or as a light color on a darker background) to indicate which of the adjacent stages of discovery is currently in progress, while the scrollable AutoDiscovery Progress Log in the lower part of the window provides detailed information on the number and type of devices being discovered. Note that the top panel also has four fields that display the model handle of the entity or device being processed.

**Figure 2-3. The AutoDiscovery Status Window**




The information displayed in the Progress Log panel is also stored in the AutoDisc.logs subdirectory of the SG-Tools directory. Individual log files are labeled ADISC.OUT and use extensions that identify the machine name, user name, date and time. For example:

```
ADISC.OUT.machineA.userB.032496.09:22:46
```

# Background Discovery

After running AutoDiscovery on a given range of IP addresses, you can continue the discovery process through the **Background Discovery** option, which is available from the Topology view's **Edit** menu for Universe, Network, and LAN models. Background Discovery runs at intervals of your choosing and will attempt to discover devices at IP addresses that could not be contacted during the AutoDiscovery session. If you have not yet run AutoDiscovery against a particular Network or LAN model, you can still use Background Discovery as long as a Network Address and Subnet Mask were

were specified and saved for that model when it was created (whether manually or via AutoDiscovery). If so, Background Discovery will generate a list of addresses that are within the range for that model, and will then attempt to contact each address. In either case, you can control the scope and frequency of Background Discovery through the dialog boxes shown in Figure 2-4 and Figure 2-5.

To configure and run Background Discovery against a particular Universe, Network, or LAN model, perform the following steps:

1. With the Topology view of the Universe, Network, or LAN model to be discovered in **Edit** mode, select **Edit > Background Discovery**. This will display the Background Discovery dialog box shown in Figure 2-4.

2. Enter a numeric value in the **Time Interval** field and use the adjacent menu button to select the desired unit (days, hours, or minutes).

3. If you want models to be created for IP addresses that could only be contacted via ICMP pings, set the **Pingables?** menu button to **Yes**.

4. Click the **IP Address** button to open the Show and Hide dialog box (Figure 2-5), which contains two scrollable lists of IP addresses. The **Undiscovered Address** list shows all the addresses that Background Discovery will attempt to contact and model. The **Excluded Address** list shows all other addresses within the applicable range.

5. Use the Filter/Search feature and arrow buttons to scan the lists and move addresses between them as desired.

6. Click **Cancel** to dismiss any changes you have made or click **OK** to confirm the current list contents and go back to the Background Discovery dialog box.

7. Click in within the **Community Names** panel, then edit to add or remove entries as needed.

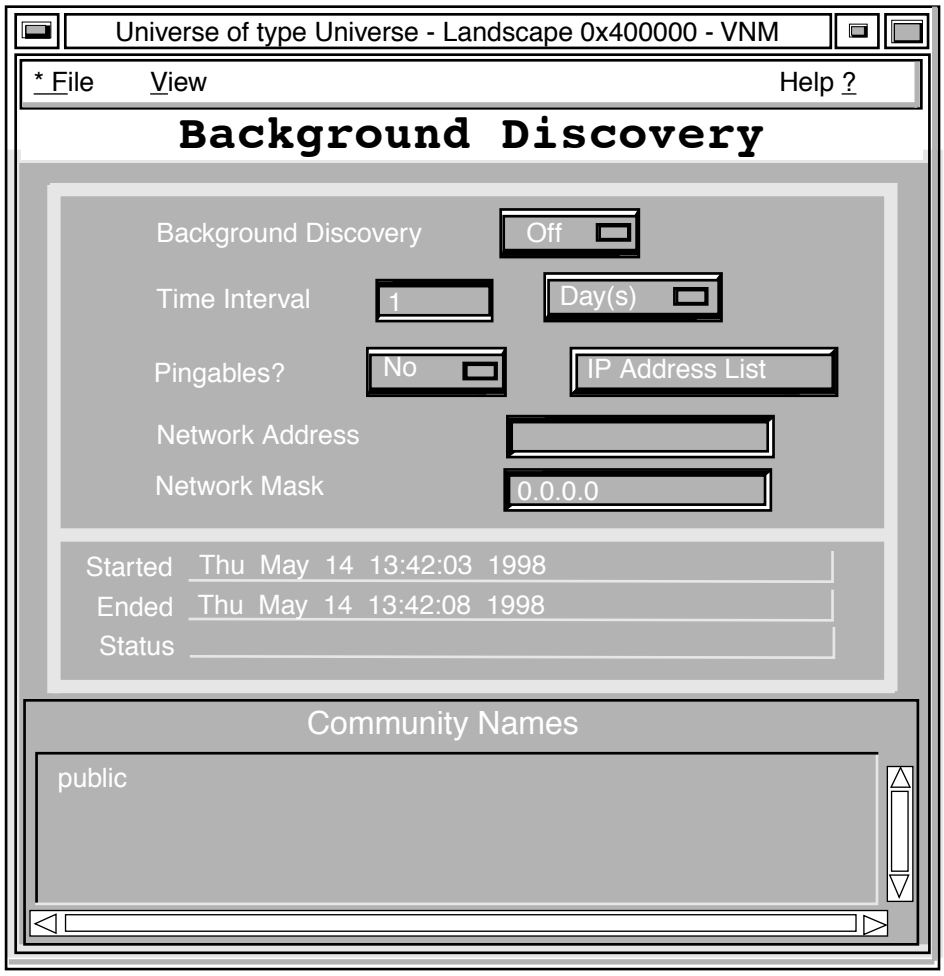
8. When you are satisfied with all settings within the dialog box, toggle the **Background Discovery** menu button to select the **On** option. (This will *enable* Background Discovery without actually starting the first session.)

9. From the **File** menu, select **Save All Changes** to start the Background Discovery session. The current date and time will be displayed in the Background Discovery dialog box's **Start** field, and the **Status** field will display the message "Running."

Once started, Background Discovery will attempt to contact each of the IP addresses in the **Undiscovered Address** list. Whenever a model is created for a particular address, the address is automatically moved to the **Excluded Address** list. The discovery session ends when all addresses on the Undiscovered list have been tried. If any addresses remain on this list at the end of the session, Background Discovery will automatically restart on behalf of this model at the time interval you have specified, unless you manually disable Background Discovery by toggling the **On/Off** button mentioned in Step 8 above. The functionality of both the Background Discovery and Show

and Hide dialog boxes is discussed in more detail immediately following the corresponding figures.

**Figure 2-4. The Background Discovery Dialog Box**




## The Background Discovery Dialog Box

Accessible from the **Edit** menu of any Universe, Network, or LAN Topology view (select the **Background Discovery** option), this dialog box lets you control the scope and frequency of Background Discovery sessions for the associated model. Dialog box components are discussed individually below.

**Background Discovery** Off / On

Toggle this menu button **On** or **Off** to enable/disable Background Discovery for the selected model. Selecting **Save All Changes** from the **File** menu will

start the discovery session only if this button is set to **On**. When no addresses remain on the Show and Hide dialog box's **Undiscovered Address** list (Figure 2-5), this button will automatically be toggled to **Off**.

**Pingables?** No/Yes

As a default, Background Discovery will not create pingable models for IP addresses that could only be contacted via ICMP pings. If you want these models to be created, toggle this menu button to the **Yes** setting.

IP Address List

Click this button to open the Show and Hide dialog box (Figure 2-5), which lets you view and modify the list of IP addresses that Background Discovery will attempt to contact.

**Network Address** 132.177.118.24

This field displays the network address of the model from which Background Discovery was invoked. If the model was created manually with no network address specified, this field will be blank, and the **Status** field (described below) will display "No addresses to discover" when you attempt to start the discovery session.

**Subnet Mask** 255.255.255.0

This field displays the subnet mask of the model from which Background Discovery was invoked. If the model was created manually with no subnet mask specified, this field will be blank, and the **Status** field (described below) will display "No addresses to discover" when you attempt to start the discovery session.

**Started** Thu May 07 14:42:03 1998

This field displays the starting date and time for the current or most recent Background Discovery session for the selected model.

**Ended** Thu May 07 14:42:48 1998

This field indicates when the most recent Background Discovery session for the selected model ended. The field is blank if there is a session in progress.

**Status** Running

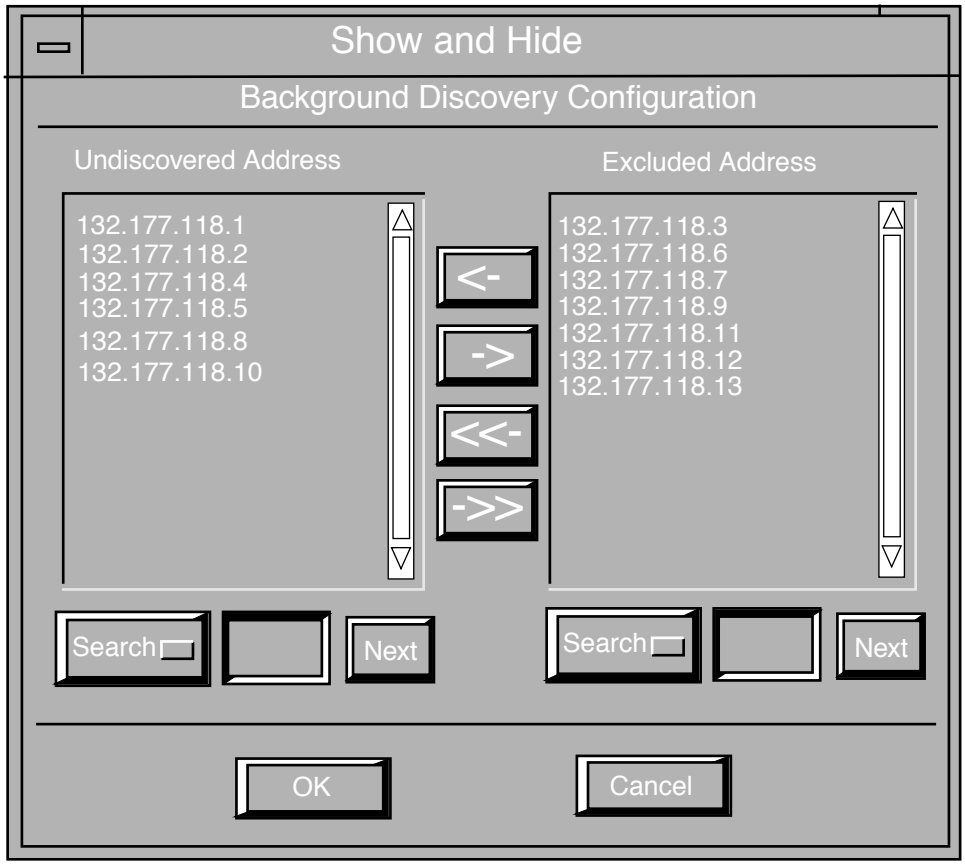
This field displays the status message "Running" when a Background Discovery session is in progress, or "No addresses to discover" if no addresses

remain on the Show and Hide dialog box's Undiscovered Address list (in which case, the Background Discovery button will be toggled to **Off**).

### Community Names

If AutoDiscovery has already been run on behalf of the selected model, this panel displays the community names used for the AutoDiscovery session. If not, the default community name "public" is displayed. You can edit this panel to add or delete names as needed for subsequent Background Discovery sessions, but your edits will not affect a session that is already in progress.

**Figure 2-5. The Show and Hide Dialog Box**




## The Show and Hide Dialog Box

Accessible from the Background Discovery dialog box's **IP Address List** button, this dialog box lets you "show and hide" IP addresses in the sense that you can determine which addresses will be included on the Undiscovered Address list that Background Discovery will attempt to contact. Components of the dialog box are discussed individually below.

### Undiscovered Address

This scrollable list shows all the IP addresses that Background Discovery will attempt to contact when next started for the selected model. As a default, the list includes all in-range addresses for which models have not yet been created, but you can also move addresses to this list from the **Excluded Address** list using the arrow buttons described below. For a given session, Background Discovery will attempt to contact each of the undiscovered addresses once. New sessions will start automatically at the interval you have specified until no addresses remain on this list.

### Excluded Address

This panel lists all other in-range IP addresses that are *not* included in the Undiscovered Address list. As a default, this list includes addresses for which models have already been created, but you can also move addresses to this list from the **Undiscovered Address** list using the arrow buttons described below.


Clicking this arrow button moves a selected IP address from the **Excluded Address** list to the **Undiscovered Address** list.


Clicking this arrow button moves a selected IP address from the **Undiscovered Address** list to the **Excluded Address** list.


Clicking this arrow button moves all IP addresses from the **Excluded Address** list to the **Undiscovered Address** list.


Clicking this arrow button moves all IP addresses from the **Undiscovered Address** list to the Excluded Address list.


Available for each of the IP address lists in this dialog box, these menu buttons provide two methods of locating a particular IP address or group of addresses. When set to **Filter**, the list above the button will display only addresses that contain whatever string you enter in the field to the right of the button. When set to **Search**, the list will display only the *first* address containing the string you have entered. You can then click the adjacent **Next** button to display the next address containing that string.


This button is used only in conjunction with the **Search** option of the **Search/ Filter** button described above. Clicking **Next** displays the next IP address that contains the string entered in the adjacent field.


Click this button to close the Show and Hide dialog box.


Click this button to cancel any changes you have maderedisplay the settings that were in effect when the dialog box was opened.

# Chapter 3

# Using AutoDiscovery

*This chapter provides a step-by-step explanation of the recommended procedure for using AutoDiscovery to create a comprehensive topological model of your network.*

## Before You Start

Although AutoDiscovery can be run on smaller networks that do not contain routers or on network sections for which the routers are not accessible, the initial discovery session for a larger network must be run at the Universe level of the SPECTRUM topology scheme and there should be **at least one router model present**. In fact, the more routers you have already modeled, the better the coverage that will be achieved by your initial discovery session. If you need to create a router model, perform the following steps:

1. Bring up SPECTRUM and navigate to the Universe-level Topology view.
2. Place the view in Edit mode and select **New Model** from the **Edit** menu. This will display the Select Model Type dialog box, which lists the available model types.
3. Select the appropriate router type by clicking on it, then click on **OK** to display a model creation dialog box.
4. In the model creation dialog box, enter a valid model name, IP address and, if applicable, an SNMP community name.
5. Click on **OK**. An icon representing the router model appears in the Universe view. The icon displays a green label once SPECTRUM makes contact with the device.

## Creating the Initial Topological Model

The recommended procedure for using AutoDiscovery is to run it initially at SPECTRUM's Universe topology level with only Router Discovery enabled, so

that only Phase One discovery will be performed. As explained in Chapter2, Phase One discovery models and places only routers, IP Class A, B, and C networks, wide area links, and LANs. It does not discover details within the discovered LANs. To have AutoDiscovery do Phase One discovery only, perform the following steps:

1. Put the Universe View into Edit mode and select the **AutoDiscover** option from the Edit menu.

2. When the main AutoDiscovery dialog box is displayed, ensure that **only** the Router Discovery method is selected (the selection button next to the method name appears in color if the method is selected). Deselect other methods as necessary by clicking on them.

3. Use the IP Address Ranges panel to establish the boundaries of your network by specifying at least one IP address range within which AutoDiscovery will operate. For each range you wish to specify, enter the low and high addresses in the IP Range field then click on the **Add** button. See the discussion of IP Address Ranges in Chapter 2 for more information.

4. Look at the Discovery Protocols panel and make sure that **SnmpPif** is selected. When the dialog box is initially opened, all four protocols are selected. You can deselect any protocols that you know are not applicable to your network; however, SnmpPif *must* be selected for Phase One (router) discovery to operate properly.

5. If you wish to use an SNMP community name in addition to the default ("public"), click in the Community Name field, then type in the name and click on the **Add** button to add it to the list of community names. Repeat as necessary for additional names, but remember that AutoDiscovery searches your network for community names in the order they are listed here. The first name a device responds to is the one that is used for the model. If your network entails multiple community names and you have a preference as to naming, be sure to list names in order of preference.

6. Click on the **Options** button (or select **Options** from the **File** menu) and check the three optional settings. The "Hierarchy" mapping option should be selected by default. Do not change the setting to "Flat" unless you want all discovered elements to be placed within a single fanout model. Change the Save Database setting to "True," if desired. It is recommended that you select one of the AutoPlace options ("Radial" or "Tree") unless you want to manually arrange the icons for discovered elements. Remember, however, that in order for the AutoPlace feature to work, you must take the topology view out of Edit mode *before* the discovery session is completed.

7. Select the **Settings** option from the main dialog box's **File** menu and then select the **Save Current** option to write the current settings to the VNM database. The saved settings will then become the default settings for subsequent AutoDiscovery sessions.

8. As noted in Step 6, take the topology view out of Edit mode if you want to use AutoPlace, then click on the main dialog box's **Start** button. This will display the AutoDiscovery Status window. When the discovery session is over, an "AutoDiscovery complete" message box will be displayed.

9. The Universe view should now contain one or more network icons (IP Class A, B, or C) as well as icons for any routers connecting these networks. (Note that the seed router(s) you started with may not remain at this level; a router connecting subnets *within* an IP Class A, B, or C network is placed within the Topology view for that network.) If you didn't select either the Radial or Tree AutoPlace options, put the Universe view in Edit mode and use the mouse to drag the icons as necessary to arrange them in whatever pattern you wish. If there are so many icons that the view appears crowded, you can cluster network models as described at the end of this chapter.

10. When the icons in the Universe view are arranged to your satisfaction, navigate into the Topology views for each of the network models and arrange the icons representing the LANs, internal routers, wide area links, and other network entities discovered during Phase One. Here, too, you may wish to cluster some of the subnets to avoid overcrowded views. Be sure to follow the instructions in the section titled "Clustering Networks" which appears at the end of this chapter.

Once you have arranged all the icons resulting from "router" discovery, the next step is to expand your network model by running AutoDiscovery for each of the LAN or other network models, this time selecting one or more of the available discovery methods so that Phase Two (bridges) and Phase Three (hubs) discovery will occur. The procedure is the same as that outlined above for the Universe level with the following exceptions:

- Instead of starting AutoDiscovery from the Universe level, navigate to the Topology view of the LAN or network model to be explored and select **AutoDiscovery** from the view's **Edit** menu.

- In the AutoDiscovery dialog box, make sure that at least one of the discovery methods is selected, but do not use the Address Resolution Table method by itself. Refer to Chapter2 for detailed descriptions of these methods.

- The IP Address Ranges panel will show a default range that encompasses all of the current LAN or network. If you wish to run AutoDiscovery for only a portion of this range, delete the default and add the smaller range. Note that any range specified at this level must be a subset of the range you established previously at the Universe level.

Repeat the procedure for each of the other LAN or Network models, using the appropriate methods, protocols, and SNMP community names.

# Additional Discovery and Configuration Tips

As noted in Chapter 1, AutoDiscovery depends on the information available from devices in your network *at the time the application is run*. If a device is temporarily out of service, it will not be modeled. Also, if the information a device contains has not been updated to reflect current configurations, then AutoDiscovery may not be able to correctly model those configurations. For these reasons, you may not discover every device in your network the first time you go through the procedure described in the previous section, even though you will have used all three discovery phases. Each additional discovery session, however, will further refine your network model. Moreover, subsequent sessions will take less time since you will have established appropriate IP address ranges as defaults.

Another reason for running additional discovery sessions is to ensure correct placement of routers if you opt to manually cluster discovered models as described below.

## Clustering Networks

If a discovery session generates a large number of IP Class A, B, or C network icons at the Universe level, or a large number of LANs within an IP network model, you may want to cluster groups of these specific network types within generic network models (Model Type = Network). This will produce a less cluttered-looking view and can make it easier to locate a given entity within the overall network mapping scheme.

A cluster can be any meaningful grouping of IP networks or LANs. For example, you might want to group all the networks associated with a specific department in your organization, or all the LANs within a given building. The only real prerequisite is that the organization scheme makes sense to you. To divide a large network into more manageable clusters, do the following:

1. Navigate to the Universe-level Topology View and decide which IP network icons you want to cluster.

2. Use the Edit menu's New Icon option to create a model of type "Network" for each cluster you've identified.

3. Use the Edit menu's Erase, Cut, and Paste options to relocate each IP network icon from the Universe level to within the Topology View for the generic network model in which you want it clustered. Begin by erasing any connection pipe icons that are attached to the IP network icon. Then cut the IP network icon from the Universe View and paste it into the Topology View for the generic network model.

   **Do NOT move any router or WA_Link models**; these will be placed correctly when you perform the next step.

4. Rerun AutoDiscovery at the Universe level. This will place routers and wide area links within the correct clusters and will regenerate connection pipe icons as necessary. Your clustering will be retained.

5. When discovery is complete, arrange the contents of the Universe and generic network-level Topology Views to your preference.

6. Examine the Topology View for each of your IP network models. If there are too many LAN models, cluster them within generic network models as explained above, then rerun AutoDiscovery for that IP network and arrange the resulting icons as desired. Again, do not move router icons; allow AutoDiscovery to place them in the proper cluster.

When you have completed the procedures outlined in this chapter, you will have an easily accessible, multilevel network model that is structured to allow full exploitation of SPECTRUM's power for effective monitoring and management. You should continue to run AutoDiscovery periodically and after known changes in your network's configuration. See Appendix B for instructions on running AutoDiscovery at predetermined intervals via the UNIX "cron" facility or the NT Schedule Service.

# Appendix A

# IP Addresses

*This appendix provides background on the structure and use of IP (Internet Protocol) addresses, which are used to define the network boundaries within which AutoDiscovery will attempt to find and model devices.*

## IP Address Structure

An IP address represents a connection to the network, not a hardware interface. The address combines a network identifier segment (net ID) with a host identifier segment (host ID). The net ID segment of an IP address is assigned by the Internet Assigned Numbers Authority (IANA). The host ID segment is determined by the network administrator.

IP addresses are expressed in "dotted quad" notation, i.e., they are made up of four 8-bit parts, separated by "dots" (periods). Each of these binary parts is written using decimal numbers and has a maximum decimal value of 255. The function of each part differs depending on network class (A, B, or C) as explained in the following sections.

### Class A Networks

The most significant bit in the first part of the address is always 0. The Net ID forms the first part and Host ID the last three parts.

| | |
|---|---|
| Net IDs range | 0 to 127 |
| Host IDs range | 0.0.0 to 255.255.254 |
| Broadcast to all hosts in net | 255.255.255 (all ones) |
| Full address range | 0.0.0.0 to 127.255.255.255 |

## Class B Networks

The two most significant bits in first part of the address are 1 and 0 respectively. For class B, the Net ID forms the first *two* parts of address and the Host ID the last two parts.

| | |
|---|---|
| Net IDs range | 128.0 to 191.255 |
| Host IDs range | 0.0 to 255.255 |
| Broadcast to all hosts in net | 255.255 (all 1's) |
| Full address range | 128.0.0.0 to 191.255.255.255 |

## Class C Networks

The two most significant bits in first part are 1 and 1. The Net ID forms the first three parts of address and the Host ID only the last part.

| | |
|---|---|
| Net IDs range from | 192.0.0 to 255.255.255 |
| Host IDs range | 0 to 254 |
| Broadcast to all hosts in net | 255 (all 1's) |
| Full address range is | 192.0.0.0 to 255.255.255.255 |

## Deciphering an IP Address

Three steps are involved in reading an IP address. For example, to read the IP address **132.177.118.24** you would do the following:

1. Convert the first decimal segment (**132**) to its binary equivalent.

   **132** (decimal) = **1 0 0 0 0 1 0 0** (binary)

2. Next, identify the class of the network. In this case, it is a **Class B** network because the first two bits of the first segment are **1** and **0,** respectively.

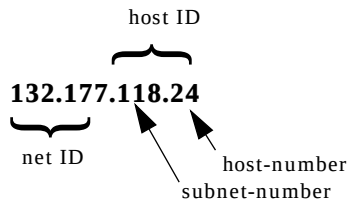
3. Identify the host portion of the address. Since this is a Class B network, the following must be true:

   **132.177** is the net ID, and **118.24** is the host ID.

# Subnets and Subnet Addresses

A single IP network can be partitioned into multiple subnets by dividing the host ID number into two parts: a subnet-number and a host-number. Figure A-1 shows the breakdown of an IP address for a Class B network.

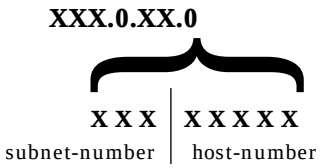
**Figure A-1. Class B Network IP Address**




When referring to a subnet, and not any particular host in the subnet, the host-number portion of the address is omitted or set to zero. For subnet number 118 in the above example, the address **134.141.118.0** refers to the full subnet.

For a Class C network, the first three parts of the IP address are used for the net ID, leaving the fourth part as the host ID. To create a subnet number in this case, you must break the 8-bits of the fourth part of the address into a subnet-number and a host-number as shown in Figure A-2, where the net ID is XXX.0.XX .

**Figure A-2. Class C Network IP Address**




This scheme provides eight subnets (0 through 7), with 32 host-numbers for each subnet (0 through 31), supporting a total of 256 addresses. Note that the breakpoint between the subnet address and the host number is determined by the system administrator. The subnet mask indicates which bits represent the network or subnet address portion of the address and which represent the host ID portion.

# Appendix B

# AutoDiscovery from the Command Line

*This appendix explains how to run AutoDiscovery from your operating system's command line, either by manual entry or by a crontab script that will automatically start the application at regular predetermined intervals.*

## Entering Startup Commands

Once you have established your network bounds and other settings by running AutoDiscovery through the user interface described in Chapters 2 and 3, you can start subsequent sessions from your operating system command line by performing the following steps:

1. Start SpectroSERVER.
2. Navigate to the `Install-Tools` directory and enter one of the following shell-specific commands:

   **`source setup.csh`** (if you are in the c shell)

   **`. setup.ksh`** (if you are in the korn or bourne shell)
3. Navigate to the `SG-Tools` directory.
4. Enter one of the following commands:

   **`autodisc -vnm <machinename> -mh <modelhandle> -n -log`**
   `(or)`
   **`autodisc -vnm <machinename> -mh <modelhandle> -config`**

   where the string ***machinename*** is the name of the machine on which SpectroSERVER (the VNM) is running, and the string ***modelhandle*** is the unique 6-digit hexadecimal number identifying the particular model against which you want to run AutoDiscovery. To determine the

modelhandle for a model in SPECTRUM, bring up the Command Line Interface and enter the command: **`show models | grep <`*`modelname`*`>`** (see the ***Command Line Interface User's Guide*** for more information). Note that if no modelhandle is specified, AutoDiscovery will run at the Universe level. Use the **-n** argument if you wish to suppress display of the AutoDiscovery dialog box. This is necessary in cases where the command will be executed as part of a crontab script (see following section). When AutoDiscovery's user interface is suppressed, the **`-log`** parameter will redirect all the progress and status messages normally displayed in the AutoDiscovery Status window (see Chapter 2) to a logfile in the AutoDisc.logs directory.

If you want the dialog box displayed so that you can reference and/or modify previously established settings, you must use the **`–config`** parameter instead. This will display the dialog box *without* the **Start** button so that you cannot inadvertently start the session.

# Using crontab Scripts

UNIX-derived operating systems allow you to define a process as a "cron job" and run it automatically at predetermined intervals. To do this you must create a **crontab** file that identifies a particular process as a cron job and specifies the times when it will be run.

Since a windowing system is required to run AutoDiscovery and the **DISPLAY** variable (normally set in your `.profile` file) must be set, AutoDiscovery is not executed directly as a cron job from a crontab file. Instead, the crontab file executes a script that first sets the environment for AutoDiscovery, then calls it as a process of the script.

Therefore, to run AutoDiscovery as a cron job, you must first use a UNIX text editor to write a simple script that sets the environment and provides the executable name and any other applicable command line arguments. The following is a sample script for a scenario in which SPECTRUM is installed in the `/usr/Spectrum` directory and AutoDiscovery is located in the `SG-Tools` directory:

```
#/bin/sh

# Run your .profile so that the DISPLAY variable is set

..profile

# Change to the SG-Tools directory, which contains
autodisc

cd /usr/Spectrum/SG-Tools

# Run autodisc
```

```
autodisc -vnm wkstn1 -n
```

Note that the command that actually starts execution (**`autodisc -vnm wkstn1 -n`**) is entered in the same format used when starting the application directly from the command line as described earlier. You can create this script anywhere you want; you tell the cron facility where to find it by the entry you make in the crontab file as described below.

Crontab files are placed in the `/var/spool/cron/crontabs` directory.

Each line in a crontab file identifies a particular UNIX process and tells where it resides and when it should be run. This information is specified in six fields separated by spaces and arranged as follows:

***minutes hours day-of-month month day-of-week command-sequence***

1. ***minutes*** indicates number of minutes. Range is 0 to 59.
2. ***hours*** indicates number of hours. Range is 0 to 23.
3. ***day-of-month*** indicates day of month. Range is 1 to 31.
4. ***month*** indicates month of year. Range is 1 to 12.
5. ***day-of-week*** indicates day of week. Range is 0 to 6. Sunday = 0
6. ***command-sequence*** indicates the command sequence required to start a process. Only the first line (up to a % character or end-of-line character) is executed by the shell.

Fields 1 through 5, which tell UNIX when to start a process, can accept input in any of the following formats:

- a one or two-digit number, for example: 11
- a series of numbers separated by commas, for example: 7,14,21
- a range of numbers separated by a dash, for example: 1-7
- an asterisk (*) character. The asterisk is either filler or a wild card. It indicates that the job will be done for all possible values in a field, unless obviated by another field; e.g., if you specify a certain day of the week, an asterisk in the day-of-month field will be ignored.

Here are some examples of entries for fields 1 through 5. Each initiates a process at the time and frequency indicated.

Daily at 10AM: **0 10 * * ***

Daily at 3PM: **0 15 * * ***

Weekly at 11:59AM on Friday: **59 11 * * 5**

Weekly at 9AM on Monday, Tuesday, and Wednesday: **0 9 * * 1-3**

Twice monthly, on the 1st and 15th: **0 0 1,15 * ***

In field 6, you specify the command sequence necessary to start a process. In the case of AutoDiscovery, you simply provide the path to the directory where the script is located and the name of the script.

Thus, to run the script you created to run AutoDiscovery as a cron job, do the following:

1. Create a crontab file using the following command:

   **crontab -e**

   This opens an existing crontab file (or creates one if none exists) using a UNIX text editor.

2. Move to an empty line and enter the time specifications for the script (fields 1-5), the path, and the script name. For example, to run the AutoDiscovery script named *adisc* at noon every Friday, the entry would appear as follows:

   **0 12 * * 5 /usr/myname/bin/adisc**

3. Save your crontab file using the appropriate command for your platform and editor. If your script and crontab file entry have been entered properly, AutoDiscovery will run automatically at the specified time using the last settings that were saved via the AutoDiscovery dialog box.

## SPECTRUM Schedule Manager

Another way to run cron jobs on Solaris systems or to schedule automatic, periodic AutoDiscovery sessions on Windows NT systems is through the SPECTRUM Control Panel's Schedule Manager feature. You can access this feature by clicking the Scheduler button in the Control Panel's Server Administration panel. The Schedule Manager provides a point-and-click interface for scheduling execution of commands and scripts. For Solaris systems, Schedule Manager automatically interprets your entries and places them in your crontab file. For Windows NT users, the Schedule Manager provides access to NT Schedule Service. For more information, refer to ***About the SPECTRUM Control Panel***.

# Appendix C

# Glossary

*This appendix is a glossary of acronyms and technical terms used in this guide.*

**ARP Table:** Address Resolution Protocol Table. Router table that resolves an IP address to a physical (MAC) address. Read by AutoDiscovery to find a list of devices with which the router has communicated.

**Community Name:** Used as a security element by the SNMP protocol. Identifies the community to which a device belongs. Default name is "public."

**Connection Pipe:** In DevTop, Cablewalk, and Topology views, logical connections are represented by "pipe" icons. These connections do not necessarily represent cables, but rather logical connections between devices.

**crontab:** A type of UNIX file that lists commands to be automatically executed at specific dates or times. A crontab script can be used to initiate AutoDiscovery sessions at predetermined intervals.

**Domain Name:** Part of naming hierarchy in an NIS environment. Identifies a particular NIS name server.

**FDDI:** Fiber Distributed Data Interface – a high-speed, dual ring, token passing network designed to run over multi-mode fiber optic cable.

**ICMP:** Internet Control Message Protocol. Part of Internet Protocol. Used to handle errors and control messages at the IP layer.

**Icon:** A graphic symbol that represents a SPECTRUM model. Icons incorporate double-click zones that provide access to views that display operating statistics and configuration data for the model.

**Inference Handler:** A portion of the C++ code that provides intelligence to the SpectroSERVER. An inference handler performs a single, specific task.

**Interface Address:** See IP Address.

**IP Address:** A 32-bit number generally represented in dotted decimal notation that is used to identify a specific host on a specific network. (See Appendix A)

**MAC_Address:** See Physical Address.

**MIB:** Management Information Base. A collection of objects that can be accessed via a network management protocol. Devices adhering to MIB versions I and II are capable of being discovered by AutoDiscovery.

**Model:** A SpectroSERVER representation of a specific network device or network group such as a specific bridge, cable, port, etc. See Model Type.

**Model Handle:** A 6-digit hexadecimal number that identifies a SPECTRUM model. The model might be a Network Group, a LAN, or a Location such as a building, room, or rack. This number appears in the title bar of each SPECTURM view.

**Model Type:** A template describing attributes, actions, and associations for construction of a software model of a device. For example, the **LAN_802_5** model type is used to create models of 802.5 LANs.

**NFS:** Network File Service. A distributed file system that allows a set of computers to cooperatively access each other's files in a transparent manner.

**NIS:** Network Information Service. A mechanism developed by Sun Microsystems to consolidate the password, hosts, and ether files from many UNIX workstations on to a few machines called servers. A server maintains the files that describe hosts and users for a particular domain.

**Null-Layer Suppression:** The suppression of the display of intermediate layers in a network hierarchy when these layers do not contain branching.

**Octet / Octet String:** An octet is an eight-bit word. An octet string is a series of octets grouped together.

**Off-Page Reference Icon:** A topology icon for a device that is related to, but not directly part of, the current view. The off-page icon usually implies a direct connection to the current topology view. The model represented by an off-page reference icon usually resides in a higher topological level.

**Physical Address:** A physical address is associated with a specific network interface (such as an interface card). Replacing the network interface card changes the physical address of that node. A physical address is six octets (48 bits). Physical addresses are generally assigned to a device at the time of its manufacture.

**Ping:** Packet Internet Groper. A program that tests whether a destination is reachable by sending an ICMP echo request and waiting for a reply.

**Protocol:** A formal description of the messages to be exchanged and rules to be followed, in order for two or more systems to exchange information.

**Redundancy:** A networking technique that reduces failures by assigning a redundant communication path. Such paths are built into the network and a means of enabling the redundant path is provided. Certain redundant paths may not be detected by AutoDiscovery.

**Relation:** A relation is a classification describing how entities relate to each other. Each relation contains a list of rules that apply the relation to model types. Relations include (but are not restricted to): Encompasses, adjacent to, Contains, Collects, HASPART, Connects to, and Monitors. See also Rules.

**Rules:** Applying a relation to specific model types creates a rule. For example, in the Contains relation, "Network group contains LAN" is a rule stating that a network group model can contain LAN models. See also Relation.

**Subnet:** A range of network addresses contained within a larger network.

**Subnet Addressing:** The splitting of the host portion of an Internet Address into two sections. The left section describes a grouping of hosts as a subnet of the IP network. The right-most section is used to assign a unique number to each host within the subnet.

**Subnet Mask:** A bit mask used to select bits from an Internet address for subnet addressing.

**Segment:** A media link connecting nodes in a network.

**SNMP:** Simple Network Management Protocol. A network management protocol used in TCP/IP-based internets.

**Universe:** Highest level of SPECTRUM Topology views.

**Unnumbered Link:** A network segment that is not assigned an IP address. Such links are not explicitly modeled by AutoDiscovery. Instead, they are represented as discrete router-to-router connections.

**Virtual Network Machine (VNM):** The intelligent software in the SpectroSERVER.

# Index

## P



## R



## S



## U



## V



## W